

Local Routing in Graphs Embedded on Surfaces of Arbitrary Genus

Maia Fraser*

November 11, 2010

Keywords: sensor networks, local routing, topological graph, homology basis, Face Routing

Abstract

We present a local routing algorithm which guarantees delivery in all connected graphs embedded on a known surface of genus g . The algorithm transports $O(g \log n)$ memory and finishes in time $O(g^2 n^2)$, where n is the size of the graph. It requires access to a homology basis for the surface. This algorithm, GFR, may be viewed as a suitable generalization of Face Routing (FR), the well-known algorithm for plane graphs, which we previously showed does *not* guarantee delivery in graphs embedded on positive genus surfaces. The problem for such surfaces is the potential presence of homologically non-trivial closed walks which may be traversed by the right-hand rule. We use an interesting mathematical property of homology bases (proven in Lemma 4.3) to show that such walks will not impede GFR. FR is at the base of most routing algorithms used in modern (2D) ad hoc networks: these algorithms all involve additional local techniques to deal with edge-crossings so FR may be applied. GFR should be viewed in the same light, as a base algorithm which could for example be tailored to sensor networks on surfaces in 3D. Currently there are no known efficient local, logarithmic memory algorithms for 3D ad hoc networks. From a theoretical point of view our work suggests that the efficiency advantages from which FR benefits are related to the codimension one nature of an embedded graph in a surface rather than the flatness of that surface (planarity).

1 Introduction

Face routing (FR) was introduced by Kranakis, Urrutia et al in 1999 in [10]. It guarantees delivery in time linear in the size of the network, using logarithmic

*Department of Computer Science, University of Chicago, 1100 East 58th Street, Chicago, IL 60637 (maia@cs.uchicago.edu).

memory, and was the first algorithm to exploit availability of position, now a common feature in many ad hoc networks. Its impact was twofold: it showed that it was possible to find *local* algorithms for ad hoc networks, as opposed to constantly re-adjusting routing tables, and it revealed the important role *position* could play in achieving this. The word “local” in this context means that the algorithm is executed by a mobile agent traveling through the network and using only local information at each node (including position) together with a “small” amount of transported memory (commonly restricted to be logarithmic in the size of the network).

More recently, in 2004, work of Reingold [14] showed universal exploration sequences (UXS’s) are constructible in logspace thus implying that it is always theoretically possible to route locally in the above sense, even without position data (transporting memory of order $O(\log n)$ as overhead in the messages)¹. However delivery time by this method is a high polynomial in the network size (not yet optimized), making it impractical for most ad hoc applications. This makes the critical challenge for routing in 3D ad hoc networks *time efficiency* and it opens the question of whether position on a surface can help improve this efficiency in 3D networks as it did in the 2D setting. We answer this question in the affirmative.

Related recent work shows that, even using geometric information, no constant memory local routing algorithm exists for 3D UBW’s that occupy a slab of thickness greater than $\frac{1}{\sqrt{2}}$ (see Durocher et al [3]), however that work does not address algorithms using logarithmic memory.

Another related result shows that no *sublinear* memory local routing algorithm exists for directed graphs, even if they are planar (Fraser et al [6]). This means no local routing algorithm exists at all in ad hoc networks in the presence of uni-directional communication links, under the common (logarithmic-memory) definition of “local” used in the ad hoc community.

In this paper we make a first step towards addressing local, logarithmic-memory routing algorithms for undirected 3D ad hoc networks, by considering those which lie on *surfaces*. In particular we are interested in ad hoc networks (for example of sensors) which may be deployed on terrains and outer surfaces of physical structures such as vehicles, buildings or other objects. We assume such surfaces are in fact oriented two-dimensional smooth manifolds which we will approximate by piecewise linear surfaces (PLS’s). We discuss this in more detail in the next section. Given a PLS (and a homology basis for it) there exists a local, logarithmic-memory, quadratic-time routing algorithm, called *generalized Face Routing* (GFR), which is correct for *all* connected graphs that are “reasonably” embedded on the surface.

By “reasonable” we mean such that any two edges and/or reference curves (connecting curve, homology curves) may intersect each other only a bounded number of times (independent of graph size). We remark that Face Routing in the plane assumes edges of the graph to be straight lines and assumes a linear or piecewise linear curve connecting sender and destination, thus

¹The author thanks Mark Braverman for his comments on this. See also [2].

achieving this intersection property. Similarly in our case we take piecewise linear (PL) curves to guarantee reasonableness.

Like Face Routing, GFR relies on the absence of edge-crossings. It is therefore, like Face Routing, a base algorithm which would require further processing to handle edge-crossings and make it suitable for real-world ad hoc networks (see [1, 11]). With appropriate assumptions on the network and surface it is possible to apply any of the standard methods for handling edge-crossings in the plane [1, 11, 3] to surfaces, however the practicality of the assumptions involved would have to be studied in a more applied paper. We welcome input from readers with specific applications in mind.

2 Terminology and Assumptions

Assume Σ_g is a piecewise linear surface (PLS) of genus g that has been triangulated. This is a natural assumption in practice (see Section 2.4 below). We recall the **genus** of a surface is defined as the maximal number of disjoint, simple closed curves which may be removed from the surface without disconnecting it.

The algorithm GFR will be executed by an agent traveling in an embedded graph G on Σ_g . We assume, as is typical in ad hoc networks, that the time of transmission is small compared with the speed at which the network changes so that one may reasonably assume the network is fixed but unknown during execution. The agent does not have access to a global representation of the graph, which may be constantly changing, but only to the ID's of its current and neighboring vertices as well their positions on Σ_g . In addition it has access to a *homology basis* \mathcal{B} for Σ_g . This will consist of closed simple curves contained in the 1-dimensional simplicial subcomplex of Σ_g .

2.1 Homology and homology bases

The reader who is not familiar with the concept of homology or homology basis is referred to Jeff Erickson's [4] for an introduction to these concepts from a computational geometry perspective or to Hatcher [7] for a more detailed mathematical treatment. We give only a minimal overview here, of **simplicial homology** for surfaces. Assume a ring R . In this paper it will be \mathbb{Q} . For $k = 0, 1, 2$, let $C_k(\Sigma_g; R)$ be the set of all formal linear combinations of k -simplices in Σ_g with coefficients in R . $C_k(\Sigma_g; R)$ is naturally a group under addition; its elements are called **k -chains**. Define the **boundary operator**

$$\delta_k : C_k(\Sigma_g; R) \rightarrow C_{k-1}(\Sigma_g; R)$$

to be the linear map which sends each oriented simplex to the sum of its oriented boundary simplices (or to zero if there are none). Those k -chains which map to zero (i.e. have no boundary) are called **k -cycles**. They form a group which we denote $Z_k = \ker \delta_k$ for $k = 1, 2$. For our case of $R = \mathbb{Q}$, connected 1-cycles with integer coefficients can be interpreted as closed unparametrized

curves which are contained in the 1-dimensional simplicial sub-complex of Σ_g . Unconnected 1-cycles with integer coefficients are formal sums of connected ones. Let $B_k = \delta_{k+1}(C_{k+1})$ denote the k -boundaries, i.e. images of the boundary operator δ_{k+1} for $k = 0, 1$. Define the first homology group $H_1(\Sigma_g; R)$ to be the quotient group Z_1/B_1 . Its elements are called **homology classes**. They are the equivalence classes for the equivalence relation “homologous to”: two 1-cycles β and β' are said to be **homologous** if their difference is a 1-boundary, i.e. β and $-\beta'$ are the two boundary components of some formal sum of 2-simplices. When $R = \mathbb{Q}$ (or any other field), $H_1(\Sigma_g; R)$ is in fact a vector space. A **homology basis** is technically a basis of this vector space and so consists of homology classes. But, in fact, one can always find a set of 1-cycles which are connected and have integer coefficients whose homology classes form a homology basis. As a result, it is common in the computational geometry literature to speak of the set of these *closed curves* (instead of their homology classes) as the homology basis. We will adopt this usage. We remark that $H_1(\Sigma_g; \mathbb{Q}) = \mathbb{Q}^{2g}$ so a homology basis for Σ_g will consist of $2g$ closed curves.

2.2 Closed curves

As mentioned above, we may interpret 1-cycles with integer coefficients as closed (oriented) unparametrized curves or linear combinations thereof. Moreover *all* closed curves used in the homological arguments of this paper in Section 4 will be oriented and simplicial, i.e. are actually connected 1-cycles with integer coefficients. Therefore, in Section 4 we will write **closed curve** as a synonym for “connected 1-cycle with integer coefficients”, and will write **simple closed curve** if the curve has no repeated vertices². Finally, when a parametrization is assumed, we will specify this and assume the orientation is compatible. See also Remark 3.4.

2.3 Planar Representation

A set of disjoint simple closed curves μ_1, \dots, μ_g on the surface Σ_g which do not disconnect Σ_g may always be completed to a homology basis

$$\mathcal{B} = \{\mu_1, \dots, \mu_g, \lambda_1, \dots, \lambda_g\}.$$

The curves μ_1, \dots, μ_g also give a convenient way of representing the surface in the plane (see Fig. 1)³. Indeed, let \mathcal{C} consist of the union of these curves. After removing \mathcal{C} we know the resulting surface is homeomorphic to a multiply punctured plane, namely some $\mathcal{U} = \mathbb{R}^2 \setminus \bigcup_{i=1}^g D_i$, where the D_i are disks. Once we assume a particular homeomorphism Φ from $\Sigma_g \setminus \mathcal{C}$ onto $\mathcal{U} \subset \mathbb{R}^2$, we can

²The latter would unfortunately be a “cycle” in standard graph theoretic terminology; our convention avoids this.

³For illustrations in the present paper we will additionally assume that $\mu_i \cap \lambda_j$ is a single point if $i = j$ and empty otherwise (such a basis can always be found), but this is not necessary to the algorithm; any homology basis will do.

map all points and curves of interest in Σ_g (that are not completely contained in the special \mathcal{C}) to corresponding points and curves in the plane. We call this a **planar representation** of the surface.

Moreover, an embedded graph G on Σ_g gives rise to $\Phi(G)$, an embedded object (points and arcs) on $\mathcal{U} \subset \mathbb{R}^2$, which we can interpret as a graph after making the identifications given by \mathcal{C} : edges will be unions of curve segments with breaks occurring at elements of \mathcal{C} . As the agent for GFR travels in G , all coordinates of neighbors it obtains will be given in terms of planar coordinates in \mathcal{U} . See Section 2.5 for more details.

We remark however that the proof of correctness that we give for GFR in Section 4 is not on the plane but rather on the PLS surface Σ_g , as this view is notationally simpler for the homological arguments involved.

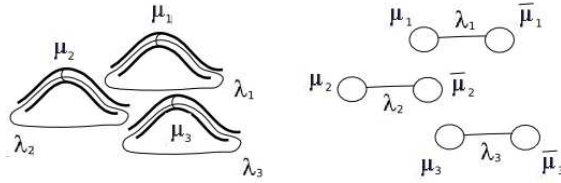


Figure 1: To the left: surface Σ_g of genus $g = 3$ viewed as the plane with 3 handles attached (after removal of one point). To the right: Σ_g viewed as the plane with 6 holes pairwise connected by the λ_i . The set $\{\mu_i, \lambda_i, : 1 \leq i \leq 3\}$ is a homology basis for Σ_g .

2.4 Note on Practical 3D Model and Future Work

The property that Σ_g be a triangulated PLS arises naturally from the standard modeling schemes for representing real surfaces. We give two examples.

Triangular Mesh Surface Models This is one of the most common modeling schemes for representing an idealized surface⁴ in computer graphics. Such a presentation consists primarily of a set of triangles (given by their vertex coordinates in \mathbb{R}^3) and may also organize this information (e.g. in so-called “triangle strips”) to facilitate stepping systematically from each triangle to a neighbor. Triangular Mesh Surfaces are in fact triangulated PLS’s. Suppose we have fixed a three dimensional ambient coordinate system in which the real world object of interest (e.g. the surface of a building) is situated. Efficient algorithms exist for generating a PLS which approximates a smooth surface using samples from the surface (see for example [12]). One then triangulates the linear pieces if they are not already triangular so as to obtain a triangular mesh model. To take this a step further, suppose one has two triangular mesh surface models, $\hat{\Sigma}_{\text{IN}}$ and $\hat{\Sigma}_{\text{OUT}}$, one on either side of the actual surface Σ_g and suitably close together. It

⁴(a triangular polytope)

is relatively straightforward to construct a piecewise linear map Φ from either $\hat{\Sigma}_{\text{IN}}$ or $\hat{\Sigma}_{\text{OUT}}$ to the plane. If these two surfaces are built “compatibly” then one can uniquely map points lying between the two surfaces to points on the plane. This allows one to convert three dimensional coordinates of points on the real world Σ_g to coordinates of points on the plane. Also, straight lines connecting nearby points on the three dimensional surface Σ_g would be mapped to PL curves in the plane and one could then use standard planar techniques to deal with edge-crossings. The aspects of “compatibility” one would require of the triangle mesh models would depend on which edge-crossing techniques one wishes to use and also the particular setting. We welcome comments from people with particular sensor network settings in mind.

Wire Frame Models Note that in structural analysis, wire frame models are often the primary objects used to model curved surfaces and an appropriate triangular mesh representation is then generated to fill out the wire frame model (see [9, 16]). For simplicity, the union of homology curves \mathcal{C}_{IN} (resp. \mathcal{C}_{OUT}) could be taken to be contained in the original wire frame model.

2.5 Data required by GFR

We now summarize GFR’s access to data. Each node has stored the model Σ_g (for example in form of a triangular mesh model) and the instructions for computing Φ (see Section 2.3). The model Σ_g may be very approximate and the corresponding data small, depending on the application. Its main purpose is to record homological information⁵. It is moreover constant: it depends *only* on the surface and not on the number of nodes or their positions. Included in this data is the homology basis \mathcal{B} . For example, assuming a triangular mesh model, this can be done by recording for every 1-simplex whether or not it belongs to the i ’th curve of \mathcal{B} (for $i \in [2g]$) and its position in a sequential numbering of the 1-simplices of that curve.

Each node can (via GPS) obtain its momentary latitude, longitude and altitude. From these it computes (using Φ) its **planar position** on $\mathcal{U} \subset \mathbb{R}^2$. The agent executing GFR can, at each node, query the neighboring nodes for their planar positions.

When the agent is started by a sender S wishing to transmit to a destination T , it computes a **connecting curve** γ . This is any PL curve in the plane connecting $\Phi(S)$ to $\Phi(T)$. γ may be mostly composed of 1-simplices, to simplify calculations.

GFR therefore operates essentially *in the plane* (within $\mathcal{U} \subset \mathbb{R}^2$) but uses identifications of certain curves in the plane and other homological data to accomplish its task.

⁵There exist various manifold reconstruction techniques from topological data analysis in recent years which yield good approximate manifolds with coinciding homological invariants.

3 Generalized Face Routing

3.1 Basic Definitions and Face Routing

Definition 3.1 (Border Walk). *A walk β in G which has no edges on its left side will be called a border walk.*

Note that border walks are boundary components of **regions**, where a region is a connected component of the complement of G in Σ_g . In general, regions in Σ_g may have multiple boundary components. We assume the standard boundary orientation convention: a boundary curve β is oriented so that the region it bounds lies to the left when β is traversed in its preferred sense. Clearly each *directed* edge forms part of the boundary of exactly one region. We thus define:

Definition 3.2 (Adjacent Border Walk). *The (at most) two border walks which share a given (undirected) edge are said to be adjacent at that edge.*

Definition 3.3 (Trivial or Non Trivial Border Walks (NTBW), Tiled Region). *If a border walk β is the only boundary component of some region, we say β is a trivial border walk, otherwise it is a non-trivial border walk (abbreviated NTBW). In other words, a closed curve is an NTBW if and only if it is not homologous to zero. We define the tiled region⁶ $\mathcal{R}(G)$ to be the union of G together with all regions of the graph which have only one oriented boundary component.*

remark 3.4. *The graph G does not in general form part of the 1-dimensional subcomplex of Σ_g . Its vertices represent mobile nodes which may lie anywhere in Σ_g and do not necessarily coincide with vertices of the fixed triangulation we have assumed for Σ_g . G should for the time being be viewed as an embedded graph on Σ_g which gives rise to the object $\Phi(G)$ in $\mathcal{U} \subset \mathbb{R}^2$. We do not specify how Φ will do this. In Section 2.4 we mentioned one way in which Φ could be defined (on a neighborhood of Σ_g in \mathbb{R}^3) so straight lines in \mathbb{R}^3 between nodes would be mapped by Φ to PL-curves in the plane. But many choices are possible depending on how one wishes to handle edge-crossings. Therefore we describe GFR in terms of the non-simplicial graph G embedded on Σ_g .*

In fact, for the proofs of Section 4, we will actually subdivide the triangulation of Σ_g so that all curves considered are simplicial. But this is not done by the algorithm; it is only a convenience for our proofs.

Regarding the (non-simplicial) definition of boundary orientation just given, we remark that it assumes the region in question has the same orientation as Σ_g . When we speak of 1-cycles bounding 2-chains, the latter may be formal sums with coefficients of any sign and 2-simplices themselves may have orientations that agree or disagree with that of Σ_g .

In keeping with standard usage, we assume the **right-hand rule** assigns as exit edge at each node the next edge clockwise from the incoming edge. Such

⁶We remark that we are departing from the definition of $\mathcal{R}(G)$ in earlier papers by the author by here including in $\mathcal{R}(G)$ any region with a connected boundary, regardless of its genus.

a routing rule then traverses a border walk containing the starting (directed) edge.

For each reference or connecting curve we assume an initial and final end-point and orient the curve from the former to the latter. For γ , these are S and T respectively, for a (closed) reference curve they both correspond to a single point on Σ_g . Moreover, we assume a simple parametrization of each curve: to each point on the curve we associate its ***t*-value**, defined as the distance from the initial point to that point *in the direction of orientation* divided by the total length of the curve. This induces a **total ordering** of the crossings of any such curve with the edges of the graph (defining higher crossings as those with higher *t*-values).

We now define a slower variant of Face Routing. We assume α to be a connecting curve or homology curve and β_0 to be a border walk that meets α (to start at S with $\alpha = \gamma$, we let β_0 be the border walk determined by the first exit edge clockwise from γ at S). In general, we start at a particular crossing $\alpha(t_0)$ of α with β_0 and *for the purposes of the algorithm only*, define **next greater** crossings in terms of *t*-values that have been shifted by subtracting $t_0 \bmod 1$; note: we do not re-define actual *t*-values, just use shifted values to order crossings so as to make $\alpha(t_0)$ the least.

Algorithm 3.5 (Modified Slow Face Routing (MSFR)). *During the entire process, stop if T is encountered.*

1. $\beta \leftarrow \beta_0$.
2. Traverse β once to find $\alpha(t_{\text{next}})$, the next greater crossing with α after the entry crossing $\alpha(t_0)$, keeping a record over all of β of the sum of crossings to the left minus crossings to the right for each homology curve.
3. If any of the total sums is non-zero, stop and exit the algorithm, otherwise travel to $\alpha(t_{\text{next}})$.
4. Set β to the adjacent border walk. $t_0 \leftarrow t_{\text{next}}$. Go to (2).

Lemma 3.6. *Let α be a connecting curve or $\alpha \in \mathcal{B}$. MSFR fully follows each component of $\alpha \cap \mathcal{R}(G)$ that it travels unless it meets T ; more precisely, if $\alpha \cap \mathcal{R}(G)$ starts at $\alpha(t_0)$ on the NTBW β_1 and ends at $\alpha(t_*)$ on the NTBW β_2 , then MSFR along α starting at $\alpha(t_0)$ on β_1 stops at β_2 or T . And MSFR is reversible (performing MSFR on the oppositely oriented curve starting at $\alpha(t_*)$ on β_2 will surely reach β_1). MSFR along all components of $\alpha \cap \mathcal{R}(G)$ takes a total time of $O(gn^2)$ and memory $O(g \log n)$, $n = |G|$.*

Although Face Routing finishes in linear time transporting logarithmic memory, it does not have the first properties mentioned (following α within $\mathcal{R}(G)$ and reversibility). This is addressed in [5]. The time bound follows from the bound d on intersections of “reasonable” curves (since MSFR will travel the boundary of each region in $\mathcal{R}(G)$ as many times as that walk’s edges meet α , the walk may have at most $O(n)$ edges and each edge may meet α at most d

times); at the same time, for each edge e traversed, MSFR must check all $2g$ curves of \mathcal{B} for crossings with e . The factor of g in the space bound is likewise due to the total sums kept for all these curves (each total can reach be recorded with $O(\log n)$ bits).

3.2 Generalized Face Routing (GFR)

Let \mathcal{L} consist of γ together with all elements of \mathcal{B} and all their oppositely oriented counterparts. Assume a fixed **indexing** of these $4g + 1$ elements *beginning with* γ .

For γ and each curve $\alpha \in \mathcal{B}$, we have already defined the **t -value** of a point on the curve; for the oppositely oriented counterparts of $\alpha \in \mathcal{B}$, we define the t -value analogously, reversing the roles of S and T .

GFR transports a list of triples (i, t_i, t'_i) , where i is the index of a curve α in \mathcal{L} , and t_i, t'_i are t -values along α .

If MSFR along some $\alpha \in \mathcal{L}$ stops before reaching T , then by Lemma 3.6 it must do so at a NTBW. If at this stage MSFR is begun on another $\alpha' \in \mathcal{L}$, it will stop at another (possibly same) NTBW. We thus hop from one NTBW to another.

Let $\Gamma = (\mathcal{N}, \mathcal{E})$ be a **virtual multi-graph** whose vertices (\mathcal{N}) are all the NTBW's of the given embedded graph G , and whose edges (\mathcal{E}) are connected *curve-like*⁷ components c of $\alpha \cap \mathcal{R}(G) : \alpha \in \mathcal{B} \cup \{\gamma\}$ where $c \subset \alpha$ is an edge between β_1 and β_2 , for $\beta_1, \beta_2 \in \mathcal{N}$ if and only if c has its endpoints at these NTBW's. An immediate consequence of this definition is:

Lemma 3.7. *Given an edge $c = (\beta_1, \beta_2)$ of Γ , which corresponds to a component of $\alpha \in \mathcal{B} \cup \{\gamma\}$, if MSFR along $\pm\alpha$ is started on β_1 at the corresponding endpoint of c then it will stop on β_2 . Hence, given any path $\beta_1, \beta_2, \dots, \beta_k$ in Γ , such that $T \notin \delta$ for any border walk δ meeting one of the $c_i = (\beta_i, \beta_{i+1}) : i = 1, \dots, k-1$, reverse MSFR can be used iteratively to return to β_1 from β_k as long as there is available a list of the t -values corresponding to the initial and final endpoints of the c_i , together with the index of the curve in \mathcal{L} to which each c_i belongs.*

Using this lemma, we define:

Algorithm 3.8 (Generalized Face Routing (GFR)). *During the entire process, exit if T is encountered.*

1. Use MSFR on γ starting from S until it stops. The agent is now at a NTBW, i.e. vertex of Γ .
2. Perform a depth first search of Γ , keeping – at each subsequent NTBW β where MSFR stops – a record of (i, t_i, t'_i) , where i is the index of the curve

⁷If α has no self-intersections then a connected *curve-like* component is just a connected component. If α does have self-intersections, we assume it is parametrized so α is a homeomorphic image of the unit circle. Then retain only that part \mathcal{P} of the circle mapped into $\mathcal{R}(G)$. Connected curve-like components of $\alpha \cap \mathcal{R}(G)$ are images of the connected components of \mathcal{P} .

$\alpha_i \in \mathcal{L}$ just followed, t_i is the t -value along α where MSFR was initiated and t'_i is the t -value along α where MSFR reached β .

remark 3.9. *That an NTBW has been previously visited is detected by GFR by the presence of a crossing $\alpha_i(t_i)$ such that (i, t_i, t'_i) and α_i correspond, as described, to one of the recorded triples.*

In the next Section, we will prove:

Proposition 3.10. *There exists a path in Γ from $\text{first}(\gamma)$ to $\text{last}(\gamma)$, resp. the first and last NTBW's met by γ . Moreover, $|\Gamma| \leq 2g$.*

This implies GFR must eventually reach $\text{last}(\gamma)$, from which MSFR will be performed on γ until T is reached, thus proving the following Theorem. Fig. 2 illustrates the process.

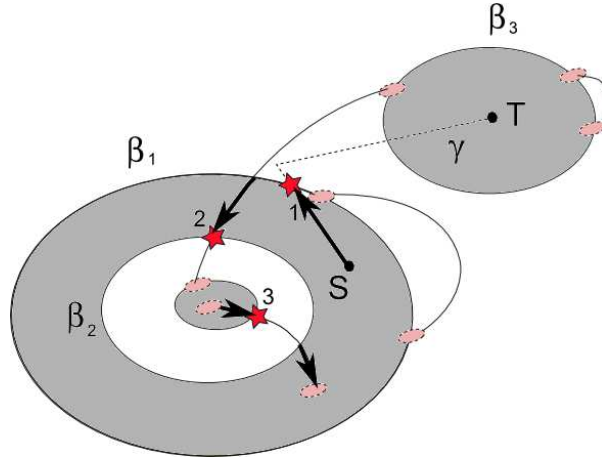


Figure 2: This is a planar representation of a surface Σ_g (with $g = 4$ in this example). The agent executing GFR sees local portions only of this picture. The shaded area is (the image under Φ of) $\mathcal{R}(G)$ - we will omit writing Φ and keep all notation from Σ . The curves $\beta_1, \beta_2, \beta_3$ are three of the NTBW's (two more are unlabeled). The dotted piecewise linear curve is γ ; i.e., the connecting curve between S and T . The pink disks are the closed disks bounded by μ_i and $-\mu_i$ respectively; the thin solid curves connecting pink discs in pairs are the λ_i (μ_i paired with $-\mu_i$). GFR on these data follows the arrows in boldface, recording a triple at each star in the numbered sequence. Note that β_3 appears in two pieces in the plane, but is in reality a single curve (when pink disk pairs are identified). Also, in going from the second star to the third, GFR will first take the arrow into the annular shaded area and then re-emerge in the central disk-like shaded area before arriving at β_3 . From β_3 the curve γ is used to reach T .

Theorem 3.11. *GFR is a local routing algorithm transporting memory of $O(g \log n)$, and guaranteeing delivery in time $O(g^2 n^2)$.*

The time bound follows since $|\Gamma| \leq 2g$ and also there are $O(g)$ reference curves in \mathcal{L} while all edges of Γ which correspond to any one of these curves can be traversed by MSFR in time $O(gn^2)$ by Proposition 3.10. The record of triples kept by GFR has size $O(g \log n)$ and traversing each edge of Γ also requires memory $O(g \log n)$ by the Lemma, hence the space bound.

4 Proof of Proposition

We now prove Proposition 3.10. It relies on a mathematical fact, Lemma 4.3, which we prove separately below. From now on we assume that the original triangulation of Σ_g has been subdivided so that all closed curves referred to – including those which are NTBWs – are *simplicial*. We do this to prove Proposition only. The algorithm *does not do this in practice*.

Proof. We divide the proof into two parts: showing $first(\gamma)$ and $last(\gamma)$ are in the same connected component of Γ , and establishing the bound on the order of Γ .

(1) As we follow γ from $first(\gamma)$ to $last(\gamma)$, let $first(\gamma) = \beta_1, \beta_2, \dots, \beta_{n-1}, \beta_n = last(\gamma)$ be the NTBWs encountered. Successive NTBWs β_i, β_{i+1} are alternately either:

- connected to each other by a component of $\gamma \cap \mathcal{R}(G)$ or are
- boundary components of a common region F .

In the first case, there is by definition an edge between β_i and β_{i+1} in Γ . In the second case, by Lemma 4.3, there is a path in Γ between them. Thus we obtain a path in Γ from $first(\gamma)$ to $last(\gamma)$.

(2) To count the NTBWs, suppose there are N of them and let \mathcal{F} be the collection of all “non-trivial” regions, meaning regions which have more than one boundary component. Let k be the number of elements in \mathcal{F} . We may cut along all but one NTBW per element of \mathcal{F} and obtain a surface Σ which is still connected, since each region that is an element of \mathcal{F} will still be attached to $\mathcal{R}(G)$, which itself is connected. We are cutting along $N - k$ curves, so $N - k \leq g$. On the other hand, cutting along just one NTBW of each element of \mathcal{F} would not disconnect the surface either, so $k \leq g$. Hence, $N \leq g + k \leq 2g$. \square

remark 4.1. *Recall that part (1) of the proof of Proposition 3.10 was divided into two cases. In the first, we considered vertices of Γ (i.e. NTBWs of G) which were adjacent in Γ since the NTBWs involved were connected by pieces of $\gamma \cap \mathcal{R}(G)$. In the second case we appealed to Lemma 4.3 to establish a path in Γ between two NTBWs which share a common region F . In fact such a path exists without using any pieces of $\gamma \cap \mathcal{R}(G)$. In other words if Γ^* is the subgraph of Γ obtained by removing all edges of the form $\gamma \cap \mathcal{R}(G)$ then the path*

of interest exists even in Γ^* . We state and prove the Lemma in this stronger form.

remark 4.2. *In the Lemma we will also allow \mathcal{B} to contain formal sums of simple closed curves, since the added generality simplifies the proof by induction.*

Lemma 4.3. ⁸ *Given a set \mathcal{B} of 1-cycles with integer coefficients generating $H_1(\Sigma_g; \mathbb{Q})$ and a connected graph G embedded in Σ_g , define Γ^* to be a virtual multi-graph, with the same vertices as Γ in Section 3.2 but with edges being connected curve-like⁷ components of $\alpha \cap \mathcal{R}(G) : \alpha \in \mathcal{B}$ only. Then whenever two NTBWs β and β' are distinct boundary components of the same region (“face”) F determined by G , there exists a path in Γ^* from β to β' .*

4.1 Intersection Number

In order to prove Lemma 4.3, we will use certain facts about homology bases expressed in terms of the notion of intersection number. We therefore define intersection number and list its relevant properties, sketching their proof in elementary terms. A formal treatment may be found in [8]. We recall that Σ_g is a simplicial complex and we use the term *closed curve* to mean a connected 1-cycle (we are using *simplicial* homology; see the brief introduction in Section 2.1). Every closed curve may be written as a sum of simple closed curves, i.e. 1-cycles with no self-intersections. Now, given any two simple closed curves α, β , we define their intersection number $\#(\alpha, \beta)$. If α, β coincide, we set $\#(\alpha, \beta)$ to zero. If they do not coincide, let σ be a shared connected 1-chain that is maximal in the sense that at its two distinct endpoints, q_1 and q_2 , α and β diverge. We assume that q_1 is the start of the shared chain and q_2 the end, using the orientation of α (which may not coincide with that of β). On the other hand, we use the orientation of β to establish a left and a right side of β (here the orientation of Σ_g is implicit as well). Now define $\iota(\sigma)$ to be zero if α is incident to the same side of β at q_1 and q_2 , and define $\iota(\sigma)$ to be $+1$ (resp. -1) if α is to the left of β at q_1 and then to the right at q_2 (resp. right and then left). Positive $\iota(\sigma)$ amounts to saying α leaves σ as an outward normal to the region bounded by β . Now, suppose this has been done for all shared sequences σ and sum $\iota(\sigma)$ over them to obtain $\#(\alpha, \beta)$. This defines intersection number for all simple closed curves. Extend this definition by bilinearity to *formal linear sums* of simple closed curves. The key properties of intersection number that we will use are:

1. $\#(\alpha, \beta)$ is a homology invariant, i.e. depends only on the homology classes $[\beta], [\alpha]$

⁸This may be stated more generally, without reference to a graph: given \mathcal{R} , a connected subsurface with boundary of a Riemann surface Σ_g with fixed choice of homology basis \mathcal{B} , and two closed curves β, β' which are boundary components of a single component of \mathcal{R}^c , there exists a sequence of homologically non-trivial curves $\beta = \beta_1, \beta_2, \dots, \beta_{n-1}, \beta_n = \beta'$ such that each pair of successive curves β_i and β_{i+1} is connected by a curve segment in the family $\{\alpha \cap \mathcal{R} : \alpha \in \mathcal{B}\}$.

2. $\#(\alpha, \beta)$ is bilinear
3. $\#(\alpha, \beta)$ is non-degenerate, i.e. any curve which is not homologous to zero must have nonzero intersection number with one of the elements of a homology basis.

These properties show that intersection number provides a non degenerate bilinear form on the first homology vector space, and they express Poincaré duality in the special case of oriented surfaces (see [17] for an accessible treatment of Poincaré duality and intersection forms). We outline a proof of the listed properties alone, giving the essential ideas.

Proof sketch. [of properties above] We now assume all curves are simple (i.e. have no self-intersections). The general case follows by straightforward extension (property 2. immediately), since arbitrary closed curves are formal linear combinations of simple ones. Moreover, for ease of discussion, we parametrize each curve compatibly with its orientation so we have an order in which to follow the curve through all its 1-simplices.

For property 1., note that if β and $\tilde{\beta}$ are homologous then β and $-\tilde{\beta}$ are the two boundary components of some region F . As mentioned in the definition, after a shared chain σ with positive $\iota(\sigma)$, α exits the region F , while after a shared chain with negative $\iota(p)$, α enters F . Since any entry (resp. exit) across β that does not have a corresponding exit (resp. entry) across β must have one across $\tilde{\beta}$, we conclude that $\#(\alpha, \beta) = \#(\alpha, \tilde{\beta})$. The argument in the first coordinate is analogous. Property 2. follows by a similar basic argument since, for example, having $\beta = \beta' + \beta''$ implies $-\beta$ and β' and β'' are the boundary components of a region F . Finally for property 3., note that if $\#(\alpha, \beta) = 0$ for all α in a homology basis \mathcal{B} , then by property 2. β would have zero intersection with *all* curves of the surface. If β were not homologous to zero then removing β from Σ_g would not disconnect Σ_g and so there would exist a path within the 1-skeleton from an edge on one side of β to an edge on the other side of β , not crossing β itself. But now, by connecting the two endpoints of this path (to β and possibly along it) one would obtain a curve which has intersection number ± 1 with β , a contradiction. \square

Definition 4.4 (Crossing). *We will refer to a maximal shared connected 1-chain of α and β as a crossing whenever α and β are on opposite sides of each other at the start and end of the chain. We will say α crosses β if such a crossing exists (even if the intersection number of the two curves is zero).*

remark 4.5. *Any 1-cycle η which has zero intersection number with a closed curve β is homologous to a sum of 1-cycles which can be parametrized (compatible with orientation) so they do not cross β . Indeed, one pairs crossings in one direction with crossings in the other and then connects a pair of crossings by two copies of a portion of β , with suitable (opposite) orientations; when taking the formal sum, these portions of β cancel out.*

On the other hand, for any 1-cycle η that has non-zero intersection number with a closed curve β there must exist a curve-like⁷ connected component c of $\eta \setminus \beta$

which starts on one side of β and ends on the other. Indeed if this were not so then following η (according to any fixed orientation-compatible parametrization) the values of ι at the crossings with β would alternate between ± 1 , thus giving $\#(\eta, \beta) = 0$, a contradiction.

4.2 Proof of Mathematical Lemma

of Lemma 4.3. We prove the Lemma by induction on the number of vertices of Γ^* . For 0 or 1 vertices it is trivially true. Now suppose Γ^* has exactly two vertices, which correspond to the NTBWs β and β' . Then these two NTBWs must bound the same region F (since an NTBW by definition is a boundary component of a region with multiple boundary components). By property 3. of intersection number there exists $\alpha \in \mathcal{B}$ such that $\#(\alpha, \beta) \neq 0$. Therefore by Remark 4.5 there exists at least one component of F . There is moreover at least one such component c . So β and β' are connected by c and hence Γ^* is connected.

Now assume the statement of the Lemma holds whenever Γ^* has k or fewer vertices. Suppose Γ^* has $k + 1$ vertices and $k \geq 2$. Let β and β' be two distinct NTBWs which bound a common region F , and let β'' be a third NTBW, distinct from β and β' . Now suppose we “cap off” β'' . More specifically, by capping off we mean cutting along β'' and gluing in a small triangulated PLS surface homeomorphic to a closed disk on either side of the cut. The resulting surface Σ' has genus $g - 1$ and it remains connected since G is (moreover, it is PLS and triangulated). The caps we glue in to β'' and $-\beta''$ are both closed disks and we will retain the name β'' for the boundary of the disk capping off β'' . Moreover, since β'' was a subgraph of G – in fact the only part of G that was affected by the cutting and gluing – let the new β'' still form the same part of G and continue to use the name G for the resulting embedded graph on Σ' . Notice that for this graph the tiled region will be extended by exactly one disk and β'' will no longer be a NTBW but all other NTBWs in Σ_g will still be NTBWs in Σ' . This is because even if β'' in Σ_g was a boundary component of F , there would still remain the two boundary components β and β' for the region after capping off and by definition of NTBW, both β and β' would still be NTBWs. If we denote the **new tiled region** in Σ' by $\mathcal{R}(G)'$, we have $\mathcal{R}(G) \subset \mathcal{R}(G)'$.

By a standard homological argument (given in Lemma 4.6 below) there exists a set \mathcal{B}' of 1-cycles with integer coefficients generating $H_1(\Sigma'; \mathbb{Q})$ such that for each $\alpha \in \mathcal{B}'$, every connected curve-like component of $\alpha \cap \mathcal{R}(G)'$ is either a connected curve-like component of $\eta \cap \mathcal{R}(G)$ for some $\eta \in \mathcal{B}$, or else a union of portions of β'' together with connected curve-like components of $\eta \cap \mathcal{R}(G)$ that meet β'' . If we define $(\Gamma^*)'$ to be the corresponding virtual multigraph, with vertices that are NTBWs in Σ' and edges that are connected curve-like components of $\alpha \cap \mathcal{R}(G)'$ for $\alpha \in \mathcal{B}'$, then we see that the vertex set of $(\Gamma^*)'$ is obtained from that of Γ^* by eliminating the single vertex β'' (and no other). Moreover (using this correspondence of vertices) the property of \mathcal{B}' given above means that any edge of $(\Gamma^*)'$ (from ν to ν' say) is either an edge of Γ^* or one can find two edges in Γ^* : one from ν to β'' , the other from β'' to ν' . Thus a path between vertices in $(\Gamma^*)'$ implies a path between those same vertices in Γ^* .

We are now done since we know by the inductive hypothesis that there is a path connecting β and β' in $(\Gamma^*)'$. \square

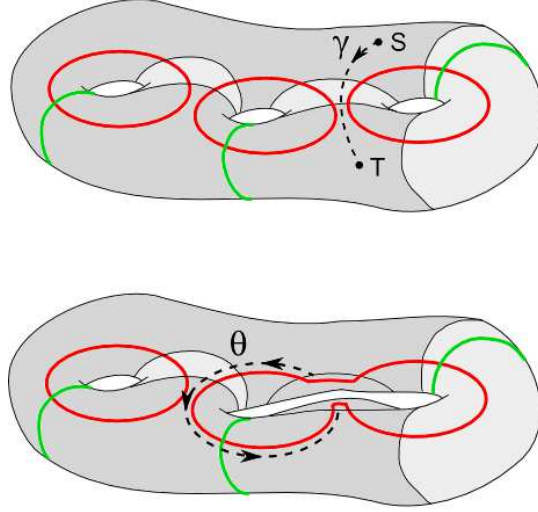


Figure 3: Induction on the genus. The top surface is Σ_g , the bottom Σ' obtained by cutting along the NTBW $first(\gamma)$ and gluing a disk in on each side. The darkly shaded area represents the tiled region in each surface. The red and green curves in the top surface are its λ_i and μ_i respectively (same surface as in Figure 1). The red curve in the middle plays the role of λ , by means of which we define the new reference curves for Σ' as shown on the bottom surface (red/green retained). If we solve the routing problem based on a connecting curve θ in the surface Σ' (which has been taken to be $\lambda \setminus first(\gamma)$ and is shown dashed) this will give us a solution to the original problem which was in terms of γ in Σ_g . But the surface Σ' has genus strictly less than g , so we are done by induction.

The argument we just made is illustrated in Fig. 3 in terms of routing. We now state and prove the result we used concerning homology bases and capping off.

Lemma 4.6. *Let \mathcal{B} be a set of 1-cycles with integer coefficients which generates $H_1(\Sigma_g; \mathbb{Q})$ and Σ' the surface obtained by capping off a closed homologically non-trivial curve β (and $-\beta$ too). Then there exists another set \mathcal{B}' of 1-cycles with integer coefficients which generates $H_1(\Sigma'; \mathbb{Q})$ such that for each $\alpha \in \mathcal{B}'$, every connected curve-like⁷ component of $\alpha \cap \mathcal{R}(G)'$ is either a connected curve-like component of $\eta \cap \mathcal{R}(G)$ for some $\eta \in \mathcal{B}$, or else a union of portions of β together with connected curve-like components of $\eta \cap \mathcal{R}(G)$ that meet β .*

Proof. We construct a set of 1-cycles which generates the homology of Σ' and has the required properties.

Since β is not homologous to zero, there is some $\alpha \in \mathcal{B}$ such that $\#(\alpha, \beta) \neq 0$ by property 3. above. Denote it λ and let $m = \#(\lambda, \beta) \neq 0$. Now consider the following set of 1-cycles with integer coefficients:

$$\mathcal{S} = \{m\alpha - \#(\alpha, \beta)\lambda : \alpha \in \mathcal{B}\}.$$

All of these curves have zero intersection number with β . Using Remark 4.5, each one is homologous to a sum of (parametrized) closed curves not crossing β . Let \mathcal{S}' be the set of all these closed curves; they are also closed curves in Σ' . We claim the elements of \mathcal{S}' generate the first homology of Σ' .

Warning: when we write $[\cdot]$ we mean a homology class in $H_1(\Sigma_g, \mathbb{Q})$; the homology classes in $H_1(\Sigma_g, \mathbb{Q})$ we will simply describe as such and not give them a short-hand notation.

Let δ be an arbitrary closed curve in Σ' . Since β bounds an open region D in Σ' that is homeomorphic to an open disk, any piece of δ which is a chord of D is homologous (in Σ') to a connected sub-chain of β . Using this to substitute all such chords of δ , we obtain that δ is homologous (in Σ') to a curve which does not cross β and in fact lies outside D . Without loss of generality we may take this new curve as δ (since we are interested in generating the homology of Σ'). It corresponds to a curve in Σ_g (by replacing 1-simplices on the boundary of the cap with 1-simplices of β , as usual). Denote this curve in Σ_g also by δ . It is homologous in Σ_g to a linear combination of the elements of \mathcal{B} , say

$$[\delta] = \sum_{\alpha \in \mathcal{B}} a(\alpha)[\alpha]$$

with coefficients $a(\alpha) \in \mathbb{Q}$ indexed by $\alpha \in \mathcal{B}$. We have

$$0 = \#(\delta, \beta) = \sum_{\alpha \in \mathcal{B}} a(\alpha)\#(\alpha, \beta).$$

Therefore,

$$\begin{aligned} \left[\sum_{\alpha \in \mathcal{B}} \frac{a(\alpha)}{m} \{m\alpha - \#(\alpha, \beta)\lambda\} \right] &= \sum_{\alpha \in \mathcal{B}} \frac{a(\alpha)}{m} [m\alpha] - \frac{1}{m} \sum_{\alpha \in \mathcal{B}} a(\alpha)\#(\alpha, \beta)[\lambda] \\ &= \sum_{\alpha \in \mathcal{B}} a(\alpha)[\alpha] = [\delta]. \end{aligned}$$

None of the elements of \mathcal{S}' crosses β and so neither does a linear combination of them. We thus have two curves, $\sum_{\alpha \in \mathcal{B}} \frac{a(\alpha)}{m} \{m\alpha - \#(\alpha, \beta)\lambda\}$ and δ which lie outside D and are homologous to each other in Σ_g . Denote the first curve η and let κ be a 2-chain in Σ_g with boundary $\eta - \delta$. When we cap off β as described earlier we produce the curve $\eta - \delta$ in Σ and a new 2-chain κ' in Σ'_g whose boundary is $\eta - \delta$. Thus η and δ are also homologous in Σ' . So \mathcal{S}' indeed generates the first homology of Σ' . \square

5 Conclusions

Face Routing (FR) is at the base of most position-based routing algorithms used in ad hoc networks today. It is a logarithmic-memory, local algorithm guaranteeing delivery in embedded graphs on the plane but not on positive genus surfaces. We have exhibited a position-based algorithm, GFR, which guarantees delivery for embedded graphs on surfaces of arbitrary genus. It is also local and uses logarithmic memory. Like FR, it is a base algorithm which as a next step could be adapted to handle edge-crossings. We welcome communication on particular settings of interest, as the method of handling edge-crossings would have to be somewhat tailored to the application. While universal exploration sequences (UXS's) provide a *non position-based* logarithmic-memory, local routing algorithm for any network, the time is a high polynomial (> 16) and no better algorithm has until now been proposed which would apply to 3D ad hoc networks. FR and GFR take linear and quadratic time respectively. By restricting to the setting of graphs on surfaces which retains the codimension-one character of graphs in the plane, we have obtained (with GFR) an FR-like gain in time efficiency for the task of local, logarithmic-memory routing in a class of 3D ad hoc network.

References

- [1] P. Boone, E. Chavez, L. Gleitzky, E. Kranakis, J. Opatrny, G. Salazar and J. Urrutia, Morelia Test: Improving the Efficiency of the Gabriel Test and Face Routing in Ad-hoc Networks, in Proc. SIROCCO 2004, R. Kralovic and O. Sykora eds., Springer Verlag, 2004, LNCS 3104, pp. 23-34.
- [2] M. Braverman, On ad hoc routing with guaranteed delivery, arxiv (2008), <http://arxiv.org/abs/0804.0862>.
- [3] S. Durocher, D. Kirkpatrick and L. Narayanan, On Routing with Guaranteed Delivery in Three-Dimensional Ad Hoc Wireless Networks, in Proc. ICDCN 2008, Springer Verlag, 2008 LNCS 4904, pp. 546-557.
- [4] J. Erickson and K. Whittlesey, Greedy optimal homotopy and homology generators, in Proc. SODA 2005, SIAM, 2005, pp. 1038-1046.
- [5] M. Fraser, Local Routing on Tori, in Proc. ADHOC-NOW 2007, Springer Verlag, 2007, LNCS 4686, pp. 153-166.
- [6] M. Fraser, E. Kranakis J. Urrutia, Memory Requirements for Local Geometric Routing and Traversal in Digraphs, in Proc. CCCG, 2008, pp. 195-198.
- [7] A. Hatcher, Algebraic Topology, Cambridge University Press, 2001.
- [8] M. Hirsch, Differential Topology, 6th ed., Springer Verlag, 1997.

- [9] R. Klein, Linear Approximation of Trimmed Surfaces, in *The Mathematics Of Surfaces VI*, Oxford University Press, 1994, pp. 209-212.
- [10] E. Kranakis, H. Singh and J. Urrutia, Compass Routing on Geometric Networks, in *Proc. CCCG*, 1999, pp. 51-54.
- [11] F. Kuhn, R. Wattenhofer and A. Zollinger, Ad-Hoc Networks Beyond Unit Disk Graphs, in *Proc. DIALM-POMC 2003*, ACM, 2003, pp. 69-78.
- [12] M. Margaliot, C. Gotsman, Piecewise-linear surface approximation from noisy scattered samples, in *Proc. IEEE conference on Visualization*, IEEE, 1994, pp. 61-68.
- [13] W.S. Massey, *Algebraic Topology: An Introduction*, Springer Verlag, 1967.
- [14] O. Reingold, Undirected ST-Connectivity in Log-Space, in *Proc. STOC 2005*, ACM, 2005, pp. 376-385.
- [15] G. Vegter, Computational Topology, in *CRC Handbook of Discrete and Computational Geometry*, J.E. Goodman and J. O'Rourke eds., CRC Press, 2004, Chapter 32, pp. 719-742.
- [16] A. Yamada, K. Shimada and T. Itoh, Energy-Minimizing Approach to Meshing Curved Wire-Frame Models, from *5th International Meshing Roundtable*, Sandia National Laboratories, 1996, pp.179-194.
- [17] G. M. Ziegler, online course notes, Topology BMS Basic Course Winter 07/08, TU Berlin, 2008, <http://www.math.tu-berlin.de/~ziegler/TOP3/notes10.pdf>.